Object Delineation as an Optimization Problem, a Connection Machine Implementation

Pascal Fua

Artificial Intelligence Center
SRI International
333 Ravenswood Avenue
Menlo Park, California 94025

Abstract

In real-world imagery, segmentation methods that rely on local image properties often fail to extract semantically meaningful features. We propose an objective function that exploits all the available photometric information. We take advantage of parallelism to effectively compute and optimize this objective function in order to find object outlines. We present our Connection Machine™ implementation and show how this technique can be used to delineate complex objects in aerial imagery and determine their elevation when using stereo pairs of images.

Introduction

In real-world imagery, object boundaries cannot be detected solely on the basis of their edge photometry because of the presence of noise and photometric anomalies. Thus, methods for delineating objects based on purely local statistical criteria are bound to make mistakes; no single parameter setting can be effective over different areas of a single image, much less for multiple images.

We address this problem by introducing "score optimizing curves" that describe objects as smooth or polygonal curves that enclose an area in the image. A global

*This research was supported in part by the Defense Advanced Research Projects Agency under contracts DCA76-85-C-0004 and MDA903-83-C-0084.

score is formed from these curves utilizing both edge information on the curve itself and the photometric information in the entire delineated area. A parallel optimization procedure deforms the curves to maximize the score and conform to object outlines.

Parallelism provides the computational power for performing the optimization in real time: at every iteration of the optimization procedure, one must recompute the photometric characteristics of the curve and its enclosed area. While this procedure could be implemented on a serial machine, the computational burden increases with the size of the enclosed area, making the optimization unacceptably slow for large objects.

Such "score optimizing curves" were originated by Terzopoulos, Kass, and Witkin as "snakes" [8, 14]. In their approach, boundaries are described as polygonal curves with a score that includes geometrical constraints and a measure of edge strength. "Snakes" do not take into account any photometric evidence outside the edge; they yield good results only if the initial position of the curve is close enough to the boundary of the object to be influenced by its edges. Because we also use *interior area* information, our curves can easily grow or shrink if the initial position is very inaccurate. By integrating more information, our algorithm also becomes more stable and less sensitive to photometric anomalies. Furthermore, in this framework, we can also take advantage of depth information and determine the elevation of an object when working with stereo pairs of images.

In this paper, we first introduce our objective function. We then describe a parallel implementation of the optimization procedure on a Connection Machine^{tm} and show how this technique can be used to delineate complex objects in aerial imagery and determine their elevation.

Objective Function

Our goal is to extract objects that conform to a particular photometric model. To discriminate among competing hypotheses, we need an objective function that measures the goodness of fit to feature models including such characteristics as area, edge, and stereo photometry, as well as shape or semantic relationships.

Because we must be able to combine corresponding measures for these models in a commensurate fashion, we choose an information-theoretic approach that enforces compatibility of the various measures. For each photometric model, we compute what we call the *effectiveness* F of the model that we define as *the difference between the number of bits needed to encode the photometry of a scene patch without the model versus with the model.* F is largest for patches that conform well to the model and can therefore be described effectively in terms of it; F measures the goodness of fit to the model. For a theoretical justification of this approach, we refer the reader to the

Minimal Description Principle introduced by Rissanen [10, 11]. Our method shares many basic concepts with the information-theory approach to segmentation described by Leclerc [9] in these proceedings; however, because our goal is to extract objects of interest rather than to segment the whole scene, many additional issues arise.

Two of the main characteristics of an object in an image are its interior texture and its contrast with the background, which produces edges. Here we explore simple models for the textured area and for the edges of an object that have proven useful in analyzing aerial imagery. The photometric evidence relevant to the edges comes from background pixels that are independent of pixels interior to regions. Therefore, these two measures are independent and we take the complete objective function F to be the sum of area and edge components, $F = F_A + F_E$. When analyzing stereo pairs of images, we also use a stereoscopic model and compute the elevation parameters of an object in the scene by optimizing the corresponding stereo effectiveness F_S.

A more robust objective function would also include a term that measures the geometric quality of a given curve and its conformity to a geometric model [3]. We have not yet incorporated such a measure in our Connection Machine implementation, and a complete discussion of a geometric term is therefore beyond the scope of this paper. However, optimizing F by itself is impractical because the "score optimizing curve" would lose its shape during the optimization. As suggested by Terzopoulos, Kass, et al. [14], we address this problem by introducing a deformation energy D that increases when the curve becomes irregular and optimizing $F - D$ instead of F; this point is discussed in detail in the implementation section.

Essential Parameters of the Objective Function

We introduce two fundamental parameters, the *scale* and the *shape coefficient*:

- **Scale.** The scale is interpretable as the unavoidable dimensional factor that converts dimensional quantities like area or length into dimensionless probabilities. Area units are thus scaled down by two powers of the dimensional unit, while boundary lengths are scaled down by a single power. The scale parameter thus controls whether or not area signature dominates edge signature.

- **Shape Coefficient.** Because we introduce the deformation energy D in our optimization, we must weigh its contribution using a shape coefficient. In our implementation

  $$D = \lambda L^2$$

  where L is the squared length of the boundary of the patch and λ the shape coefficient. D is a smoothing term required to enforce regularity of the boundaries because F is a highly nonconvex function that would be difficult to optimize by itself; λ controls the amount of smoothing.

We know of no *a priori* way to determine the scale and shape coefficient, because they characterize the fundamental balance of influences that must be specified for each application. Nevertheless, our approach provides a clear way to justify and understand the essential roles of these two parameters in feature extraction.

Area Model for Homogeneous Regions

We model a homogeneous region with area A, such as a building roof, as a planar intensity surface with a Gaussian distribution of deviations from the plane, plus anomalous pixels whose values lie outside the peak of the distribution.

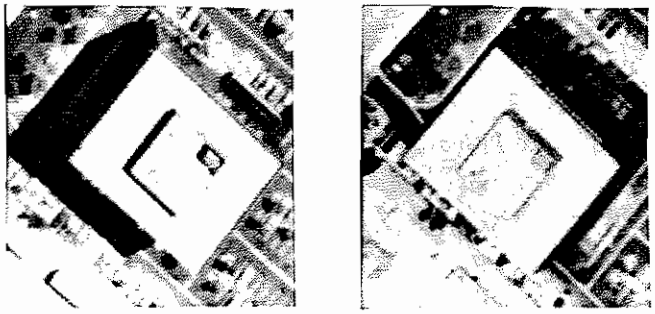


Figure 1: A stereo pair of images containing a large building

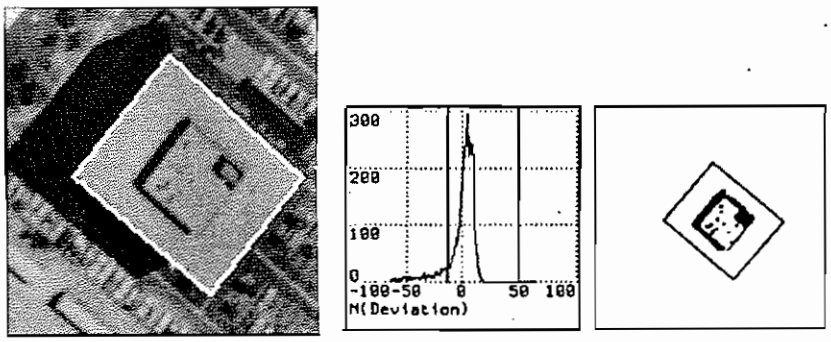
Figure 1 shows a stereo pair of images, Figure 2 a the outline of the main rooftop in the left image, and Figure 2 b the corresponding histogram of deviations from the planar fit to the intensity surface along with the left and right bounds of the main Gaussian peak. In Figure 2 c, the solid white area indicates the location of the pixels within the peak. Black areas within the outline lie outside the peak and are considered anomalous.

In an 8-bit image, it would take $8A$ bits to encode the pixel values if we did not take advantage of dependencies among pixels. Similarly it would take $k_A A$ bits to encode the same information using our region model, where

$$k_A A = n(\log \sigma + c) + 8\overline{n} - \left[n \log \frac{n}{A} + \overline{n} \log \frac{\overline{n}}{A} \right]. \qquad (1)$$

Here σ is the variance of the Gaussian distribution, n is the number of pixels in the Gaussian, and $\overline{n} = A - n$, and $c = \frac{1}{2}\log(2\pi e)$*. Note that in the computation of the

*All logarithms in this paper are base 2 logarithms.



(a) (b) (c)

Figure 2: (a) Outline of the main rooftop in the left image of Figure 1 (b) Histogram of deviations from the planar fit with left and right bounds of the Gaussian peak. (c) The solid black areas within the contour indicate the location of the pixels that do not belong to the main Gaussian peak and are considered anomalous.

encoding cost, we have not included the cost of encoding the internal parameters of the model, such as the slopes and intercept of the plane. It can be shown [10, 13] that these costs are proportional to the logarithm of the area A and are therefore very small compared to $k_A A$.

We weight all areas and lengths using the *scale parameter* s, so that the area effectiveness becomes

$$\begin{aligned} F_A &= \text{bits(area without model)} - \text{bits(area with model)} \\ &= (8 - k_A)\frac{A}{s^2}. \end{aligned} \tag{2}$$

Effect of anomaly discounting. In the left-hand graphs in Figure 3, we plot the area effectiveness F_A as a function of the radius of a square-shaped patch at the center of the images shown in the left column: a good but noisy synthetic image of a square, the same image with edge jitter, and with gross area anomalies. When we compare the results obtained *after discounting anomalies* (solid lines) with those found without anomaly discounting (dotted lines), we see that anomaly discounting can easily be *entirely responsible* for generating the local extrema (i.e., the desired shape) perceived by human observers. This is potentially a critical factor in the practical application of this approach because, as we see in Figure 2, real images nearly always have significant anomalous components.

Parallel computation of the score. The score can be efficiently evaluated on a Connection Machine because the computation only involves fitting a plane over a patch, computing the deviations histogram, and finding for the Gaussian peak the left and right bounds that yield the best value of F_A. The planar fit requires a small number of parallel summations, and the histogram can be computed in one parallel operation. All possible choices of the left and right bounds of the peak are then evaluated simultaneously, and only the best are retained.

Edge Model

We adopt the definition [2, 7, 12] of edge pixels as maxima of the local image derivative. To enforce this criterion in our information theoretic framework, we propose the following scheme.

We take the edge gradient to be $g = (\partial I/\partial x)^2 + (\partial I/\partial y)^2$ where I is the image intensity. Assuming that g ranges between 0 and M, it would take $logM$ bits per pixel to encode gradient intensities of boundary pixels in the absence of a model. The gradient of boundary pixels is expected to be higher than that of other pixels; we model this fact by describing the edge strength in terms of a vocabulary that favors high gradients. We assume that a pixel with gradient g can be described using $-\log(g/c)$ bits where $c = M^2/2$ is a normalizing constant (g/c is a probability density that must sum to 1 over all possible values of g).

We then weight all lengths by the scale factor s and estimate the edge effectiveness to be, for a boundary of length L,

$$\begin{aligned} F_E &= \text{bits(edge without model)} - \text{bits(edge with model)} \\ &= \frac{L}{s}\log M + \frac{1}{s}\sum \log \frac{g}{c} \\ &= \frac{1}{s}\sum \log \frac{g}{\gamma} \end{aligned} \tag{3}$$

where $\gamma = M/2$ and $\sum$ represents a summation over the boundary pixels.

In practice γ is computed as a percentage of the edge strength and treated as a threshold value for the edge strength under which $\log(g/\gamma)$ is taken to be 0. It can be shown [4] that all the points along a curve that maximizes F_E are maxima of the edge gradient in the direction normal to the curve and therefore satisfy our definition of an edge pixel.

The right-hand graphs in Figure 3 show the edge effectiveness of the boundaries of the square patches discussed in the previous section, as a function of their radius. In Figure 4, we plot the area and edge scores (with anomaly discounting) as a function of size when square patches (solid lines) are compared to circular ones (dotted lines), as applied to the images of Figure 3. In the case of the perfect square, the edge

score clearly provides excellent discriminating power. However, in the case of the square with edge jitter, the optimum of the edge effectiveness is much less distinct; the combination of edge effectiveness and area effectiveness has more discriminating power than either alone. Also note that the differences in effectiveness between the squares and circles are much less marked in the noisy image than in the noise-free one. This is an intuitively satisfactory behavior because the square shape is much less perceptible in the noisy image.

Parallel computation of the score. The image gradients can be precomputed using Gaussian convolution operators. The computation of the score then reduces to a global summation of the gradient intensities over the boundary pixels, which can be achieved in one parallel step.

Stereography

The simplest stereo model assumes that corresponding pixels have the same grey-levels in both images [1]. In practice, one finds deviations from this model that we encode again as a Gaussian distribution, excluding anomalies arising from such causes as occluding structures.

As in the area-encoding case, we can now determine the number of bits required to encode the area in the second image by histogramming the deviations of the intensities from their predicted values. We also want to take into account the edge quality of the contour in the second image and its edge effectiveness.

We therefore take the stereographic effectiveness term F_S to be the sum of an edge and area term:

$$\begin{aligned} F_S &= F_{AS} + F_{ES} \\ F_{AS} &= (8 - k_2)\frac{A_2}{s^2} \\ F_{ES} &= \frac{1}{s}\sum \log \frac{g}{\gamma} \end{aligned} \tag{4}$$

where A_2 is the area of the projected patch in the second image, k_2 is the average number of bits/pixel needed to encode the deviation histogram, and g the edge gradient in the second image.

We can use the effectiveness measure Eq. 5 to optimize the elevation parameters of a two-dimensional delineation found in the first image. The search space is extremely constrained because the projected shape is known and the only degree of freedom is epipolar motion in the second image.

Let us consider the stereo pair of images shown in Figure 1 and the rooftop outlined in Figure 5a. Assuming that it is horizontal, we plot in Figure 5b the value of F_S

as a function of the assumed disparity between the outline in the left image and the outline in the right image. We note that F_S presents a sharp peak for the correct match shown in Figure 5c.

Implementation and Applications

Deformable Models in Two Dimensions

To find local maxima of the objective function $F - D$, where $F = F_E + F_A$ and $D = \lambda L^2$ is the deformation energy introduced in Section , we describe object contours as deformable closed curves defined by an ordered list of contiguous points C represented by the vector X of their integer x coordinates and the vector Y of their y coordinates. During each iteration of the optimization procedure described below, X and Y are updated. C is then recomputed by drawing scan lines between points that are not contiguous anymore and merging points that have identical coordinates, thereby generating new vectors X and Y. The edge effectiveness F_E is computed using those new boundary pixels and the area effectiveness F_A of the pixels enclosed by the boundary but not belonging to it. In this way the contour can shrink or expand as required to optimize the objective function.

At every iteration, we compute the derivative of the F with respect to deformations of the contour C:

$$\begin{aligned} \frac{\partial F}{\partial X} &= \frac{\partial F_A}{\partial X} + \frac{\partial F_E}{\partial X} \\ \frac{\partial F}{\partial Y} &= \frac{\partial F_A}{\partial Y} + \frac{\partial F_E}{\partial Y} \end{aligned} .$$

In the appendix we derive expressions for these derivatives and show that they can be easily evaluated on a Connection Machine.

To perform the optimization we could use a simple gradient descent technique, but it would be extremely slow for curves with a large number of points. Instead, we modify the standard gradient procedure in two ways:

1. **Treat C as a physical system.** As in the work by Terzoupolus [14], we consider C as a "snake" that is, a physical curve defined by the vector (X, Y), embedded in a medium of viscosity $\alpha = 1/\lambda$, and moving under the influence of the potential $V = L^2 - \alpha F$. L^2, the square length of the boundary, can be computed as:

$$L^2 = \frac{1}{2} XKX + \frac{1}{2} YKY \tag{5}$$

where K is the tridiagonal matrix with coefficients $-1, 2, -1$. At every iteration of the optimization, we then solve the equation of dynamics:

$$\frac{\partial V}{\partial C} + \alpha \frac{dC}{dt} = 0 \tag{6}$$

where $\partial V/\partial C$ is the vector $(\partial V/\partial X, \partial V/\partial Y)$. Because the deformation energy L^2 in Eq. 5 is quadratic, its derivatives with respect to X and Y are linear. Thus, each iteration of the optimization amounts to solving the two linear equations:

$$\begin{aligned} KX_t + \alpha(X_t - X_{t-1}) &= \alpha \left.\frac{\partial F}{\partial X}\right|_{C_{t-1}} \\ KY_t + \alpha(Y_t - Y_{t-1}) &= \alpha \left.\frac{\partial F}{\partial Y}\right|_{C_{t-1}} . \end{aligned} \tag{7}$$

Letting $M = (I + \frac{1}{\alpha}K)^{-1}$, Eq. 7 can be rewritten as:

$$\begin{aligned} X_t &= M(X_{t-1} + \left.\frac{\partial F}{\partial X}\right|_{C_{t-1}}) \\ Y_t &= M(Y_{t-1} + \left.\frac{\partial F}{\partial Y}\right|_{C_{t-1}}) . \end{aligned} \tag{8}$$

For α large enough (typically $\alpha > .01$), the matrix M can be approximated with excellent accuracy by an n-diagonal matrix. We can therefore solve Eq. 8 simultaneously for X and Y by convolving the right-hand terms $X + \partial F/\partial X$ and $Y + \partial F/\partial Y$ with the appropriate mask. In this formulation, the value of α determines the width of the mask and how much X and Y are smoothed – the smaller α, the more smoothing.

It is worth noting that approximately the same result can be achieved by a faster although slightly less accurate procedure. Instead of solving the equations of the dynamics, we can increment X and Y by $\partial F/\partial X$ and $\partial F/\partial Y$ as in a standard gradient procedure and then recursively smooth the resulting coordinate vectors using the mask $[.25, .5, .25]$. This procedure is fast because it can be implemented using only integer additions and left shifts but no floating point operations or multiplications. In practice, the results produced by these two procedures are almost indistinguishable. The results presented in this paper have been generated using recursive smoothing; at each iteration, the X and Y vectors are convolved 10 times with the mask $[.25, .5, .25]$ except in the example shown later in Figure 7d.

2. **Normalize the derivatives of the score.** The magnitude of the derivatives is not related to the current distance of the contour to its optimal location. Therefore, for every iteration, we pick a step size and retain only the sign of the

derivatives that indicates in which direction the contour should move, resulting in a string FX of -1,0 and 1 for the X coordinates and a string FY for the Y coordinates. We then normalize the string so that $(\|FX\|^2 + \|FY\|^2)/n = \delta^2$, where n is the number of boundary points and δ the current step size, and replace $\partial F/\partial X$ and $\partial F/\partial Y$ by FX and FY in Eq. 8. This ensures that the displacement of each point is on the average of magnitude δ.

Because of the presence of the linear terms in the dynamics equation (Eq. 6), deformations are propagated along the whole curve at every iteration, making this procedure considerably faster than ordinary gradient descent.

Because the objective function is highly nonconvex, after each iteration we recompute the score and verify that it has increased. If, instead, it has decreased, the curve is reset to its previous position and the step size reduced.

The optimization proceeds until the curve stabilizes. For example, going from the initial estimates of the closed curve shown in Figure 6a to the final result shown in Figure 6d took only 10 iterations. Figure 6b and 6c show the position of the curve after three and five iterations respectively.

We now turn to the aerial image in Figure 7a. The four initial contours shown in Figure 7b yield, after optimization, the final outlines of Figure 7c. Note that the corners of the house are slightly rounded due to the presence of the smoothing term. To delineate the house more accurately, we can reoptimize the corresponding curve using less smoothing, generating the result shown in Figure 7d.

Timing considerations. As long as there are fewer points in the "score optimizing curve" C than there are Connection Machine processors, most of the computation can be performed with a virtual processor ratio of 1, with the possible exception of the planar fit and the computation of the deviation histogram required for estimating F_A, which must refer directly to the image. In the following table, we indicate the average times required to perform the various operations when dealing with either a a 64×64 image or a 128×128 image on the 8K machine we are using.

Time in seconds to:	64×64	128×128
Compute F_A	.165	.185
Compute F_E	.011	.011
Compute $\partial F/\partial X$ and $\partial F/\partial Y$	.055	.055
Update X and Y	.180	.195
Perform a complete iteration	.42	.45

If we were using a full 64K machine, the times would be the same for 256×256 and 512×512 images respectively. Note that these times are independent of the

length of the contour or its interior area. If this algorithm were implemented on a serial computer, the time required to compute F_E would grow as the length of the boundary, and the time required to compute F_A would grow as the area, which would slow down the algorithm unacceptably for any large object.

Polygonal Models in Two and Three Dimensions

The "score optimizing curves" described in the previous section behave like rubber bands that attempt to shrink-wrap the contours of an object and yield a smoothed outline. When attempting to extract polygonal objects, we can explicitly include a polygonal constraint by fitting line segments to the curve after each iteration of the optimization procedure.

We now consider the left image of the stereo pair shown in Figure 1. In Figure 8a we show three initial polygonal contours, and in Figure 8b the result of the optimization assuming that the number of vertices in the contours does not change. In the presence of corners, the polygonal constraint yields better results, provided that the location of the polygon vertices can be computed. In practice, this can be achieved by first performing the optimization with a simple rubber band, finding the high curvature points, and using these as candidate vertices for a polygonal "score optimizing curve".

After the contour outlines shown in Figure 8b are found, their elevation can be determined by optimizing the value of the stereo effectiveness from Eq. 5. Assuming that the rooftops are planes, the matching contours in the right image are shown in Figure 8c. These contours and their elevation can then be fed to a system such as the SRI cartographic modeling system [5, 6] to generate synthetic three-dimensional views of the scene.

Conclusion

We have presented, for automatically outlining object boundaries, a technique that integrates area, edge, and stereographic information with geometric models, given a very rough initial estimate of the boundary. The constraints are incorporated by defining, for curves, an objective function that is maximal when the models are satisfied exactly. The initial estimate is used as the starting point for finding a local maximum of this objective function by embedding the initial curve in a viscous medium and solving the equations of dynamics.

The strength of this "score maximizing curve" approach is that all the available photometric information is taken into account simultaneously with geometric constraints.

Parallelism is essential for a successful implementation of this technique because it provides the computational power required to perform the optimization in real time. We plan to apply this technique to investigate more sophisticated constraints, including more sophisticated geometric models than the one described in this paper, and to better understand their relevance to the feature extraction problem. Our Connection Machine implementation and our optimization scheme will allow us to quickly experiment with such constraints on numerous examples and decide their value. Such an investigation would be impractical without the possibility of performing such experiments rapidly.

This technique can also be used for semiautomated data acquisition: a photo-interpreter provides a very rough estimate of the location of an object and lets the computer determine the object's precise outline and elevation. In future work, therefore, another goal will be to provide the user with means of interactively guiding the optimization when necessary and to introduce geometric constraints that objects of interest must satisfy.

Appendix: Derivatives of the Effectiveness

Derivatives of the Area Term

To estimate the derivatives of F_A, we first compute the contribution dF_A of every point (x, y) in the image when added to the patch defined by C. As shown in Eq.2

$$\begin{aligned} F_A &= (8 - k_A)\frac{A}{s^2} \quad \text{where :} \\ k_A &= n(\log\sigma + c) + 8\overline{n} + \left[n\log\frac{n}{A} + \overline{n}\log\frac{\overline{n}}{A}\right] \\ c &= \frac{1}{2}\log(2\pi e)\,, \end{aligned}$$

which we can rewrite as:

$$k_A = n(c_1 - \frac{\log v}{2}) + n\log n + \overline{n}\log\overline{n} - A\log A,$$

where $c_1 = 8.0 - c$ and $v = \sigma^2$. To evaluate the contribution of an individual pixel we must distinguish two different cases:

1. The pixel belongs to the main gaussian peak if its deviation from the planar fit d is between the left and right bounds defined in section . In that case, n and A must be incremented by 1 while the the overall variance v is modified by

$dv \approx (d^2 - v)/n$. Therefore dF_A can be computed as follows:

$$\begin{aligned} dF_A &= (c_1 - \frac{\log v}{2}) - \frac{c_2}{2} n \frac{dv}{v} + \log n - \log A \\ &= (c_1 - \frac{\log v}{2}) - \frac{c_2}{2}(\frac{d^2}{v} - 1) + \log n - \log A \end{aligned}$$

where $c_2 = \log_e 2$.

2. The pixel does not belong to the main peak, its contribution to $\overline{n}$ and dF_A can be taken as:

$$dF_A = \log \overline{n} - \log A.$$

Having computed dF_A, we can now estimate $\partial F_A/\partial X$ using finite differences. Let us consider a boundary point $P = (x, y)$. Our implementation assumes that the boundary points themselves do not belong to the patch. There are four possible patterns for the 3×1 horizontal neighborhood centered around P:

a:	1	×	0
b:	0	×	1
c:	1	×	1
d:	0	×	0

where 0 represents a point that does not belong to the patch and 1 represents a point that does.

- Case a: If P moves to the right, the center point is added to the patch and F_A becomes $F_A + dF_A(x, y)$; conversely if P moves to the left, the left point is removed from the patch and the F_A becomes $F_A - dF_A(x-1, y)$. $\partial F_A/\partial x$ is therefore estimated to be:

$$\frac{\partial F_A}{\partial x} = + \frac{dF_A(x, y) + dF_A(x-1, y)}{2}.$$

- Case b: Similarly,

$$\frac{\partial F_A}{\partial x} = - \frac{dF_A(x, y) + dF_A(x+1, y)}{2}.$$

- Case c and d: The boundary is locally horizontal,

$$\frac{\partial F_A}{\partial x} = 0.$$

$\partial F_A/\partial X$ is the vector of the $\partial F_A/\partial x$ for all the points in C. $\partial F_A/\partial Y$ is computed similarly by replacing horizontal neighborhoods by vertical ones. Note that dF_A can be computed on a pixel-per-pixel basis and therefore in parallel for all pixels in the image. The computation of $\partial F_A/\partial X$ and $\partial F_A/\partial Y$ involves only communication with nearest horizontal and vertical neighbors — operations that are very fast on a Connection Machine^{tm}.

Derivatives of the Edge Term

We have seen in Eq. 3 that F_E is computed as

$$F_E = \frac{1}{s} \sum_{C(x,y)} \log \frac{g(x,y)}{\gamma}.$$

In practice we precompute, once and for all, the quantity Γ defined by

$$\Gamma(x,y) = \begin{cases} \log(g(x,y)/\gamma) & \text{if } s > \gamma \\ 0 & \text{otherwise} \end{cases}.$$

We also precompute the derivative of Γ, $\partial\Gamma/\partial x$ and $\partial\Gamma/\partial y$. At each iteration, $\partial F_E/\partial X$ and $\partial F_E/\partial Y$ are simply the vectors whose components are the values of $\partial\Gamma/\partial x$ and $\partial\Gamma/\partial y$ at the current boundary points.

References

[1] Barnard, S.T., "Stochastic Stereo Matching Over Scale," *Proceedings of the DARPA Image Understanding Workshop,* Boston, MA, pp.769–778, April 1988.

[2] Canny, J., "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 8(6), pp.679–698, 1986.

[3] Fua, P.V., and A.J. Hanson, "Extracting Generic Shapes Using Model-Driven Optimization," *Proceedings of the DARPA Image Understanding Workshop,* Boston, MA, pp.994–1004, April 1988.

[4] Fua, P.V., and Y.G. Leclerc, "Model Driven Edge Detection," to be published in the *Journal of Machine Vision and Applications,* 1989.

[5] Hanson, A.J., A.P. Pentland, and L.H. Quam, "Design of a Prototype Interactive Cartographic Display and Analysis Environment," *Proceedings of the Image Understanding Workshop,* pp.475–482, February 1987.

[6] Hanson, A.J., and L. Quam, "Overview of the SRI Cartographic Modeling Environment," in *Proceedings of the Image Understanding Workshop,* Boston, MA, pp.576–582, April 1988.

[7] Haralick, R.M., "Digital Step Edges from Zero Crossings of Second Directional Derivatives," *IEEE Transactions on Pattern Analysis and Machine Vision,* Vol. 6(1), pp.58–68, 1984.

[8] Kass, M., A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision,* Vol. 1(4), pp.321–331, 1988.

[9] Leclerc, Y.G., "Segmentation Via Minimal-Length Encoding on the Connection Machine," *Proceedings. of the Fourth International Conference on Supercomputing,* Santa Clara, CA, April–May 1989.

[10] Rissanen, J., "A Universal Prior for Integers and Estimation by Minimum Description Length," *The Annals of Statistics* Vol. 5, pp.416–431, 1983.

[11] Rissanen, J., "Minimum-Description-Length Principle," in *Encyclopedia of Statistical Sciences*, Vol. 5, pp.523-527, 1987.

[12] Rosenfeld, A., "A Nonlinear Edge Detection Technique," *Proceedings of the IEEE,* Vol. 58, pp.814–816, 1970.

[13] Schwarz, G., "Estimating the Dimension of a Model," *The Annals of Statistics,* Vol. 6, pp.461–464, 1978.

[14] Terzopoulos, D., "On Matching Deformable Models to Images," *Topical Meeting on Machine Vision,* Technical Digest Series, Optical Society of America, Washington, DC, Vol. 12, pp.160–167, 1987.

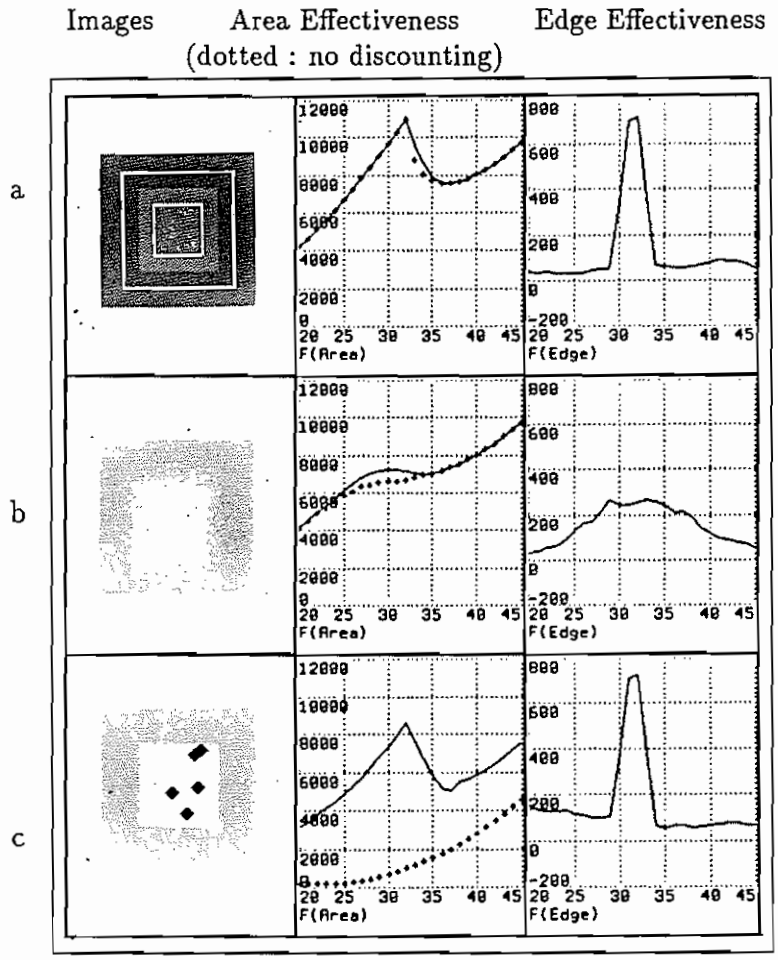


Figure 3: Area and edge effectiveness of a squared patch as a function of their radius. The patches of radius 20 and 45 are outlined on the top image.

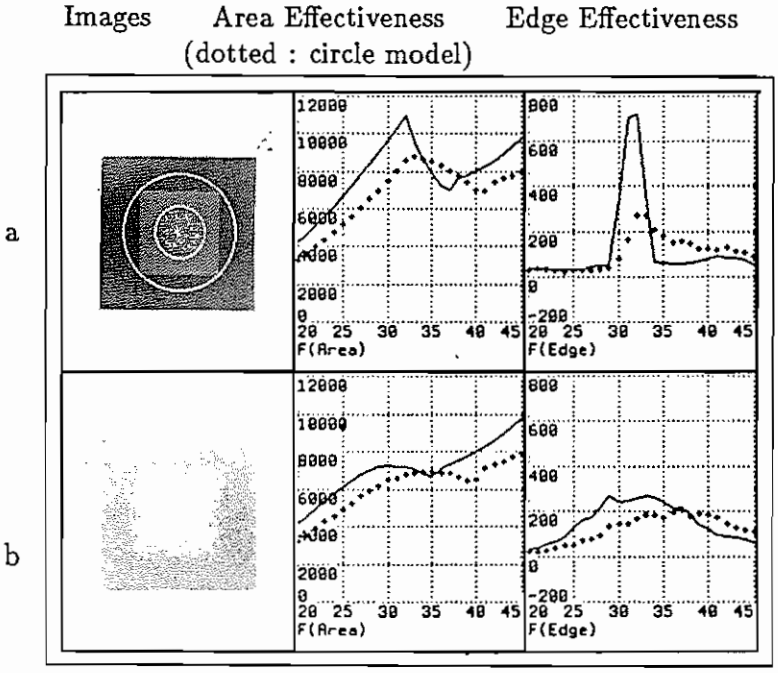
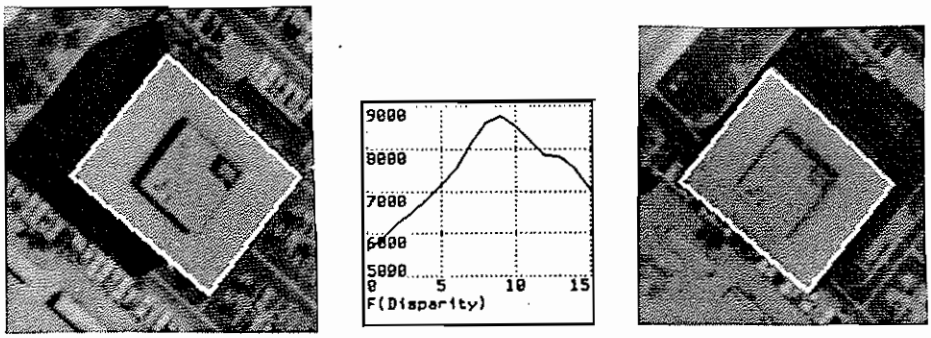


Figure 4: Area and edge effectiveness of a squared patch compared to that of a circular patch as a function of their radius. The circular patches of radius 20 and 45 are outlined on the top image.



(a) (b) (c)

Figure 5: (a) The main rooftop in the left image of Figure 1 (b) F_S as a function of the assumed disparity between left and right image. (c) The projection of the contour in the right image using the best disparity value.

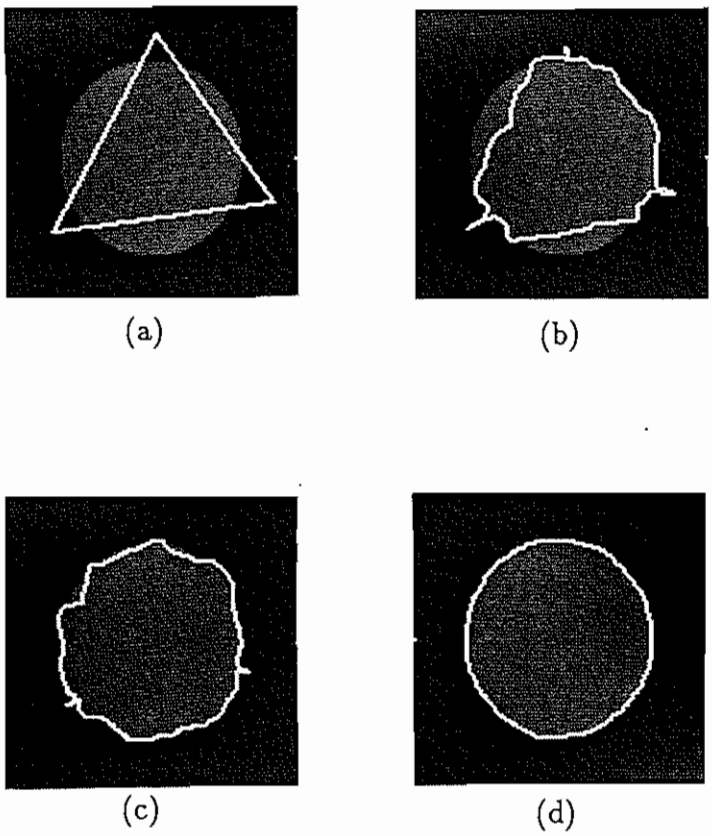


Figure 6: (a) A synthetic image of a circle and the initial position of the curve. (b) (c) The position of the curve after three and seven iterations, respectively. (d) The final outline.

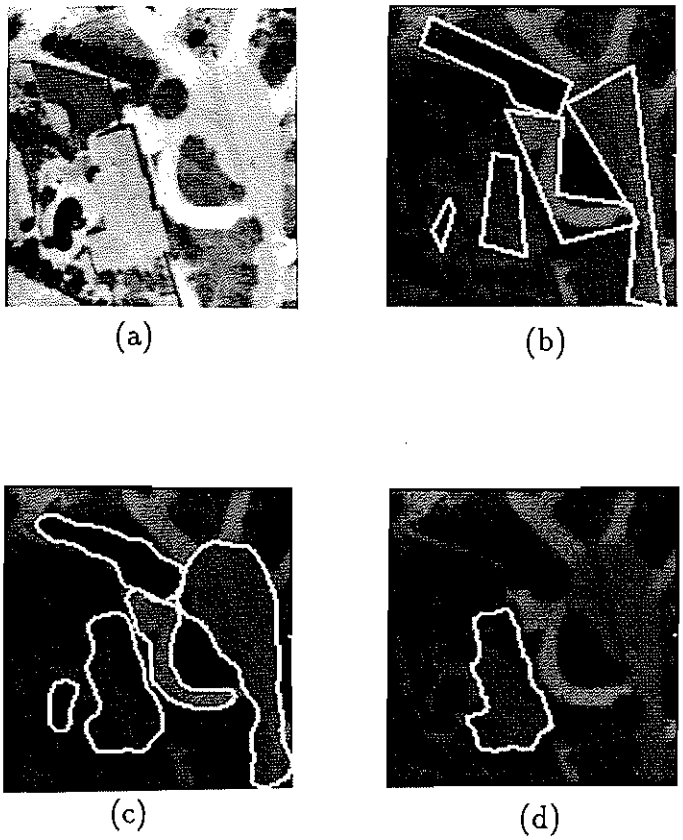
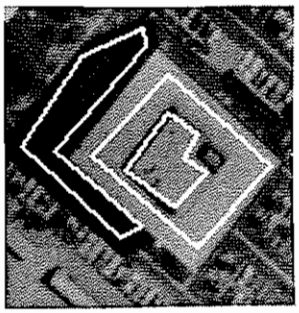
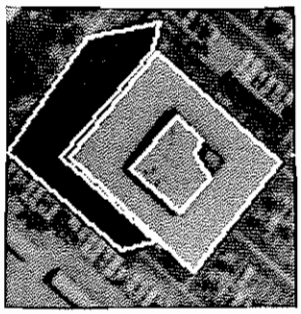


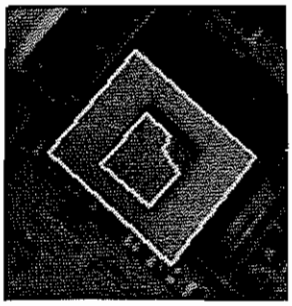
Figure 7: (a) An aerial image of a suburban scene. (b) Interactively entered initial contours. (c) Final outlines after optimization. (d) Outline of the house after reoptimization with less smoothing.



(a)



(b)



(c)

Figure 8: (a) Initial contours in the left image of the pair from Figure 1. (b) Final polygonal outlines after optimization. (c) Matching outlines in the right image.